Chicago Mafia Report

Nathan Clingman, Andrew Couch, Carter Hansen, Jason Shanker
University of Iowa

Table of Contents

Dataset Overview

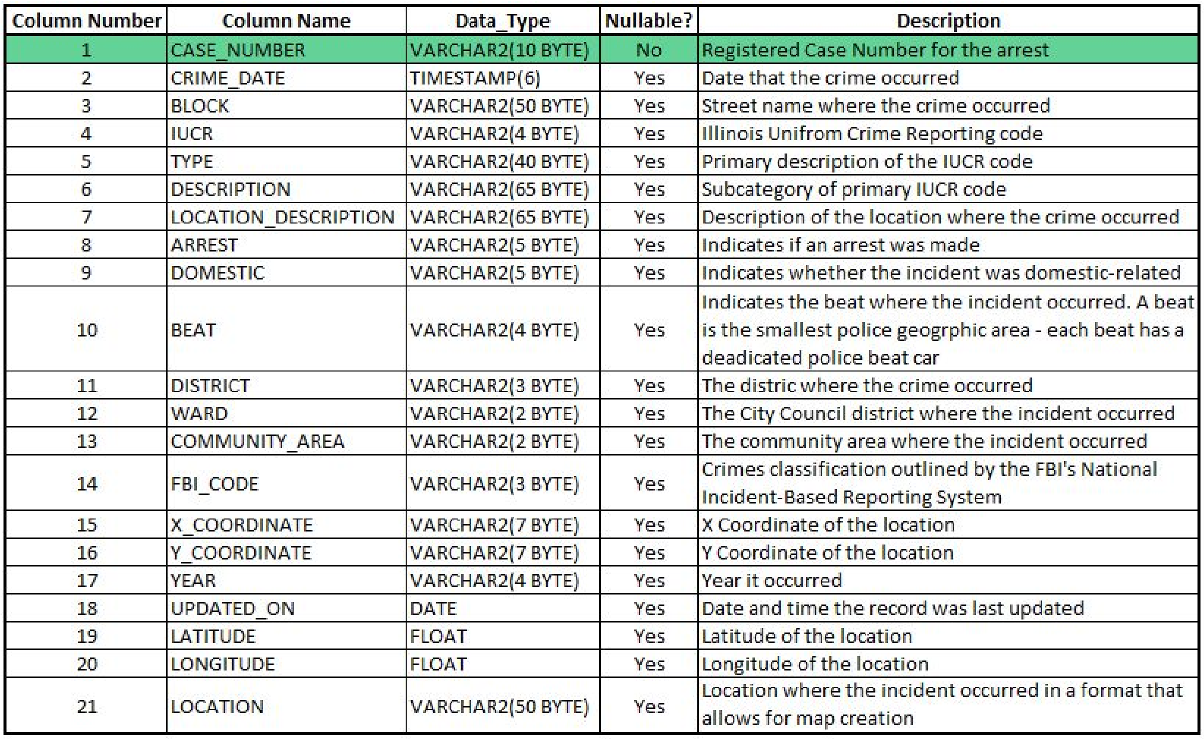
The Chicago Crimes database is a table of crimes that have occurred in Chicago, IL from 2001 to present, minus the most recent 7 days. This table includes the time, location, the type of crime, and a description of the crime committed. It has 6,928,228 rows and 21 columns. The data includes 36 unique types of crimes, over 60,000 unique blocks, 507 unique crime descriptions, 210 unique location descriptions, and 24 unique districts. There are 66,959 crimes with missing locations.

Data Preparation

Listed below is the schema used to load the data into SQL Developer. The primary key is CASE_NUMBER because the ID field contains duplicates.

```
CREATE TABLE CHICAGO
( CASE_NUMBER VARCHAR2(10 BYTE) NOT NULL
, CRIME_DATE TIMESTAMP(6)
, BLOCK VARCHAR2(50 BYTE)
, IUCR VARCHAR2(4 BYTE)
, TYPE VARCHAR2(40 BYTE)
, DESCRIPTION VARCHAR2(65 BYTE)
, LOCATION_DESCRIPTION VARCHAR2(65 BYTE)
, ARREST VARCHAR2(5 BYTE)
, DOMESTIC VARCHAR2(5 BYTE)
, BEAT VARCHAR2(4 BYTE)
, DISTRICT VARCHAR2(3 BYTE)
, WARD VARCHAR2(2 BYTE)
, COMMUNITY_AREA VARCHAR2(2 BYTE)
, FBI_CODE VARCHAR2(3 BYTE)
, X_COORDINATE VARCHAR2(7 BYTE)
, Y_COORDINATE VARCHAR2(7 BYTE)
, YEAR VARCHAR2(4 BYTE)
, UPDATED_ON DATE
, LATITUDE FLOAT(13)
, LONGITUDE FLOAT(13)
, LOCATION VARCHAR2(50 BYTE)
, CONSTRAINT CHICAGO_PK PRIMARY KEY (CASE_NUMBER));
```

Schema and Description



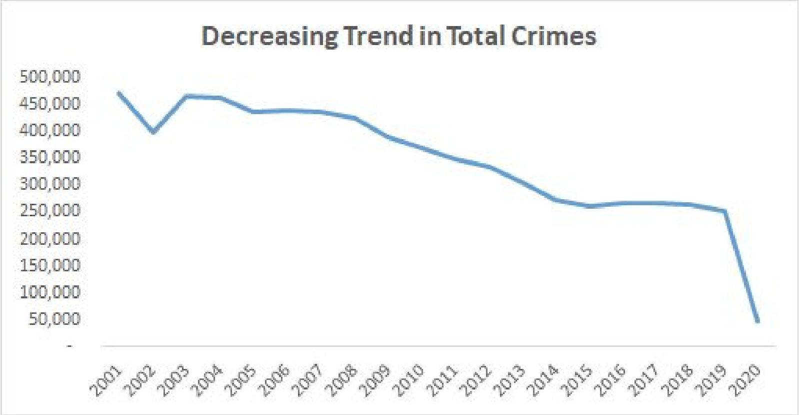
Column Number	Column Name	Data_Type	Nullable?	Description
1	CASE_NUMBER	VARCHAR2(10 BYTE)	No	Registered Case Number for the arrest
2	CRIME_DATE	TIMESTAMP(6)	Yes	Date that the crime occurred
3	BLOCK	VARCHAR2(50 BYTE)	Yes	Street name where the crime occurred
4	IUCR	VARCHAR2(4 BYTE)	Yes	Illinois Unifrom Crime Reporting code
5	TYPE	VARCHAR2(40 BYTE)	Yes	Primary description of the IUCR code
6	DESCRIPTION	VARCHAR2(65 BYTE)	Yes	Subcategory of primary IUCR code
7	LOCATION_DESCRIPTION	VARCHAR2(65 BYTE)	Yes	Description of the location where the crime occurred
8	ARREST	VARCHAR2(5 BYTE)	Yes	Indicates if an arrest was made
9	DOMESTIC	VARCHAR2(5 BYTE)	Yes	Indicates whether the incident was domestic-related
10	BEAT	VARCHAR2(4 BYTE)	Yes	Indicates the beat where the incident occurred. A beat is the smallest police geogrphic area - each beat has a deadicated police beat car
11	DISTRICT	VARCHAR2(3 BYTE)	Yes	The distric where the crime occurred
12	WARD	VARCHAR2(2 BYTE)	Yes	The City Council district where the incident occurred
13	COMMUNITY_AREA	VARCHAR2(2 BYTE)	Yes	The community area where the incident occurred
14	FBI_CODE	VARCHAR2(3 BYTE)	Yes	Crimes classification outlined by the FBI's National Incident-Based Reporting System
15	X_COORDINATE	VARCHAR2(7 BYTE)	Yes	X Coordinate of the location
16	Y_COORDINATE	VARCHAR2(7 BYTE)	Yes	Y Coordinate of the location
17	YEAR	VARCHAR2(4 BYTE)	Yes	Year it occurred
18	UPDATED_ON	DATE	Yes	Date and time the record was last updated
19	LATITUDE	FLOAT	Yes	Latitude of the location
20	LONGITUDE	FLOAT	Yes	Longitude of the location
21	LOCATION	VARCHAR2(50 BYTE)	Yes	Location where the incident occurred in a format that allows for map creation

Data Statistics

Exploratory analysis was conducted to gain insights on key individual column statistics. The key columns explored were arrest, domestic, type, location and date. Completing this analysis increased our understanding of the dataset.

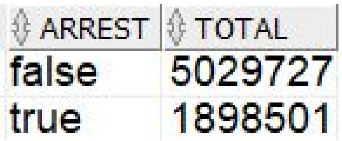
To see a general trend in crime numbers across our dataset, a count of every crime in each year found a decreasing trend in total crimes since 2001. The query below counts the total number of cases that occur.

```
SELECT YEAR,COUNT(YEAR)
FROM CHICAGO
GROUP BY YEAR
ORDER BY YEAR ASC;
```



Next the arrest variable was examined. This variable indicates true if the crime resulted in an arrest. 27.4% of all the crimes listed led to an arrest.

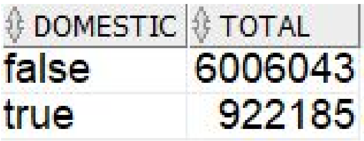
```
SELECT ARREST, COUNT(ARREST) AS TOTAL
FROM CHICAGO
GROUP BY ARREST;
```



ARREST	TOTAL
false	5029727
true	1898501

The domestic variable uses true and false to indicate if the incident is domestic related (this is defined by the Illinois Domestic Violence Act). The data shows 13.3% of Chicago crimes are domestic.

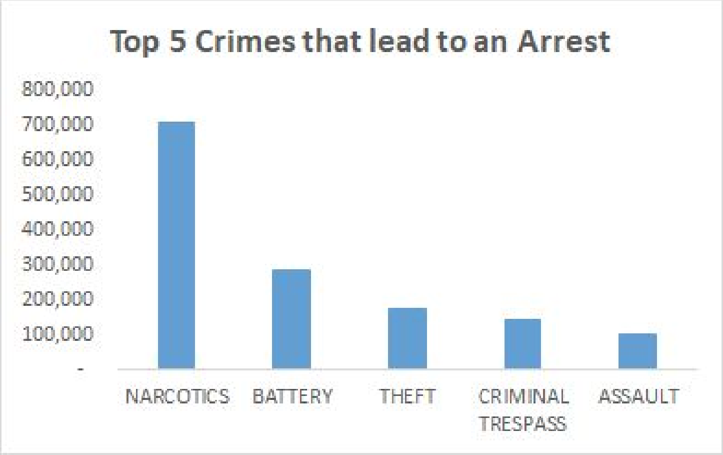
```
SELECT DOMESTIC, COUNT(DOMESTIC) AS TOTAL
FROM CHICAGO
GROUP BY DOMESTIC;
```



DOMESTIC	TOTAL
false	6006043
true	922185

To find out what crimes resulted in an arrest, the top five crimes were analyzed. Narcotics was first followed by battery, theft, criminal trespass, and assault.

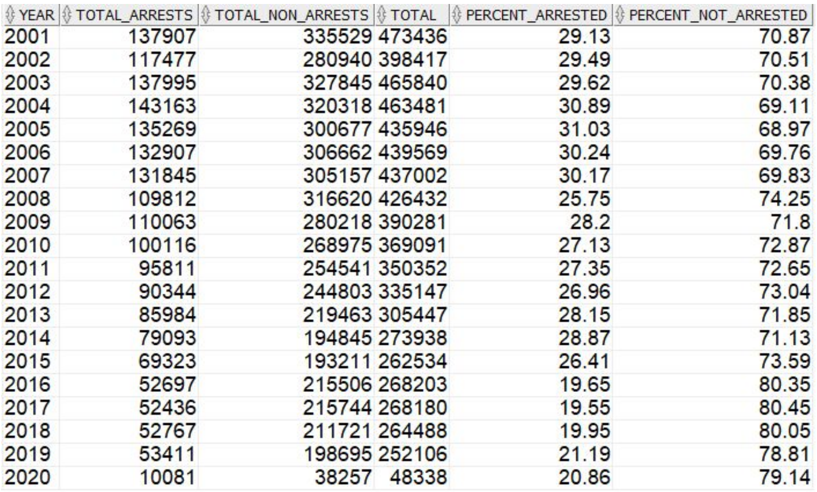
```
SELECT*
FROM (SELECT TYPE, COUNT(ARREST) AS TOTAL_ARRESTS
FROM CHICAGO
WHERE ARREST='TRUE'
GROUP BY TYPE
ORDER BY TOTAL_ARRESTS DESC)
WHERE ROWNUM <= 5;
```



The dataset contained two columns, crimes and arrest, crimes noted when an incident occured and arrest marked if the person was taken into custody. The graph shows the top 5 arrest categories from the dataset. 2004 had the most amount of arrests, where 2001 had the most crimes that were not arrested. 2005 had the highest percentage of crimes end in an arrest with 31.03%. 2017 had the least with 19.55%.

```
SELECT YEAR,SUM( CASE WHEN ARREST LIKE '%TRUE%' THEN 1 ELSE 0 END)
AS TOTAL_ARRESTS,
SUM(CASE WHEN ARREST LIKE '%FALSE%' THEN 1 ELSE 0 END) AS
TOTAL_NON_ARRESTS, ROUND(SUM( CASE WHEN ARREST LIKE '%TRUE%'
THEN 1 ELSE 0 END)+SUM(CASE WHEN ARREST LIKE '%FALSE%' THEN 1 ELSE
0 END),2) AS TOTAL,ROUND(SUM( CASE WHEN ARREST LIKE '%TRUE%' THEN 1
ELSE 0 END)/(SUM( CASE WHEN ARREST LIKE '%TRUE%' THEN 1 ELSE 0
END)+SUM(CASE WHEN ARREST LIKE '%FALSE%' THEN 1 ELSE 0 END))*100,2)
AS PERCENT_ARRESTED,ROUND(SUM(CASE WHEN ARREST LIKE '%FALSE%'
THEN 1 ELSE 0 END)/(SUM( CASE WHEN ARREST LIKE '%TRUE%' THEN 1 ELSE 0
END)+SUM(CASE WHEN ARREST LIKE '%FALSE%' THEN 1 ELSE 0 END))*100,2)
AS PERCENT_NOT_ARRESTED
FROM CHICAGO
GROUP BY YEAR
```

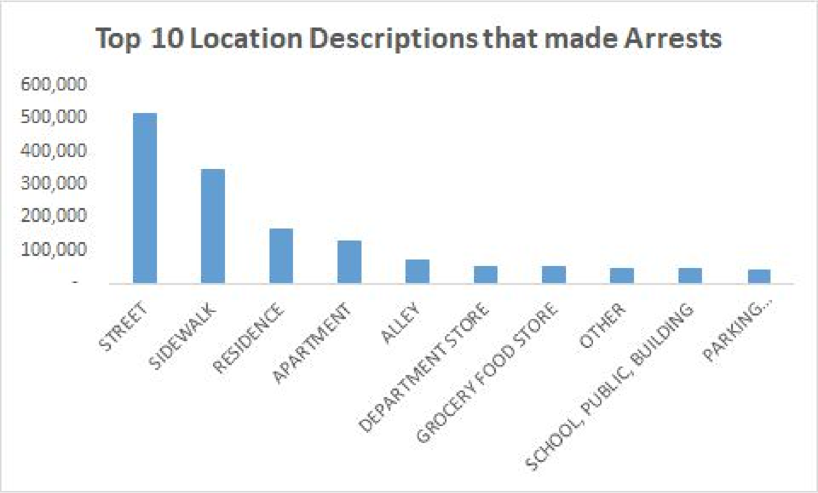
```
ORDER BY YEAR ASC;
```



YEAR	TOTAL_ARRESTS	TOTAL_NON_ARRESTS	TOTAL	PERCENT_ARRESTED	PERCENT_NOT_ARRESTED
2001	137907	335529	473436	29.13	70.87
2002	117477	280940	398417	29.49	70.51
2003	137995	327845	465840	29.62	70.38
2004	143163	320318	463481	30.89	69.11
2005	135269	300677	435946	31.03	68.97
2006	132907	306662	439569	30.24	69.76
2007	131845	305157	437002	30.17	69.83
2008	109812	316620	426432	25.75	74.25
2009	110063	280218	390281	28.2	71.8
2010	100116	268975	369091	27.13	72.87
2011	95811	254541	350352	27.35	72.65
2012	90344	244803	335147	26.96	73.04
2013	85984	219463	305447	28.15	71.85
2014	79093	194845	273938	28.87	71.13
2015	69323	193211	262534	26.41	73.59
2016	52697	215506	268203	19.65	80.35
2017	52436	215744	268180	19.55	80.45
2018	52767	211721	264488	19.95	80.05
2019	53411	198695	252106	21.19	78.81
2020	10081	38257	48338	20.86	79.14

Lastly, location descriptions that had the most number of arrests were examined. Street and sidewalk are the top 2 locations, followed by residence and apartment.

```
SELECT * FROM (
SELECT LOCATION_DESCRIPTION,COUNT(ARREST) AS TOTAL_ARREST
FROM CHICAGO
WHERE ARREST='TRUE'
GROUP BY LOCATION_DESCRIPTION
ORDER BY TOTAL_ARREST DESC)
WHERE ROWNUM <=10;
```



Data Quality Issues and Resolution

Issues

Using ID as the primary key in the original dataset caused duplicates.

Resolution

Since case_number was used as primary key beforehand this did not cause any duplicates in our data table. Null values from case_number were dropped which only had 4 missing values for.

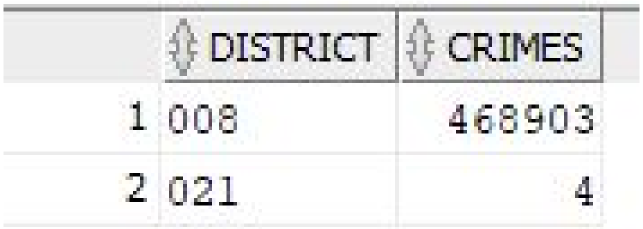
```
CREATE TABLE CHICAGO AS ( SELECT CASE_NUMBER, CRIME_DATE, BLOCK,
IUCR, TYPE, DESCRIPTION, LOCATION_DESCRIPTION, ARREST, DOMESTIC,
BEAT, DISTRICT, WARD, COMMUNITY_AREA, FBI_CODE, X_COORDINATE,
Y_COORDINATE, YEAR, UPDATED_ON, LATITUDE, LONGITUDE, LOCATION
FROM CHICAGO WHERE CASE_NUMBER IS NOT NULL GROUP BY
CASE_NUMBER, CRIME_DATE, BLOCK, IUCR, TYPE, DESCRIPTION,
LOCATION_DESCRIPTION, ARREST, DOMESTIC, BEAT, DISTRICT, WARD,
COMMUNITY_AREA, FBI_CODE, X_COORDINATE, Y_COORDINATE, YEAR,
UPDATED_ON, LATITUDE, LONGITUDE, LOCATION );
```

Questions and Output

Question 1:

For question 1, the most dangerous and safest districts in Chicago were found. Dangerous and safest districts were created by using two subqueries and combining the results into a compound query. The subqueries count the crimes for each district and filters the districts that have the maximum and minimum amount of crimes. Then the results are combined together using a union clause.

```
SELECT DISTRICT, COUNT(*) AS CRIMES
FROM CHICAGO
GROUP BY DISTRICT
HAVING COUNT(*) = (
   SELECT MAX(CRIMES)
   FROM(
      SELECT DISTRICT, COUNT(*) AS CRIMES
      FROM CHICAGO
      GROUP BY DISTRICT))
UNION
SELECT DISTRICT, COUNT(*) AS CRIMES
FROM CHICAGO
GROUP BY DISTRICT
HAVING COUNT(*) = (
   SELECT MIN(CRIMES)
   FROM(
      SELECT DISTRICT, COUNT(*) AS CRIMES
      FROM CHICAGO
      GROUP BY DISTRICT));
```



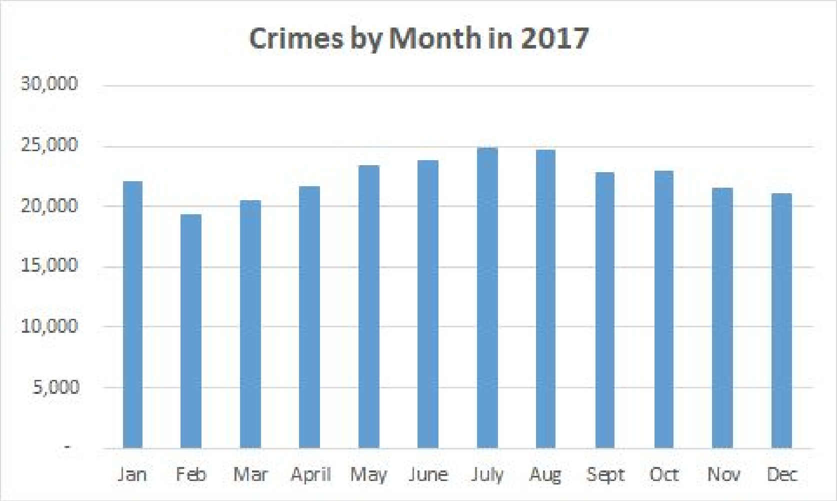
	DISTRICT	CRIMES
1	008	468903
2	021	4

District 008 has the most amount of crimes and district 021 has the least amount of crimes.

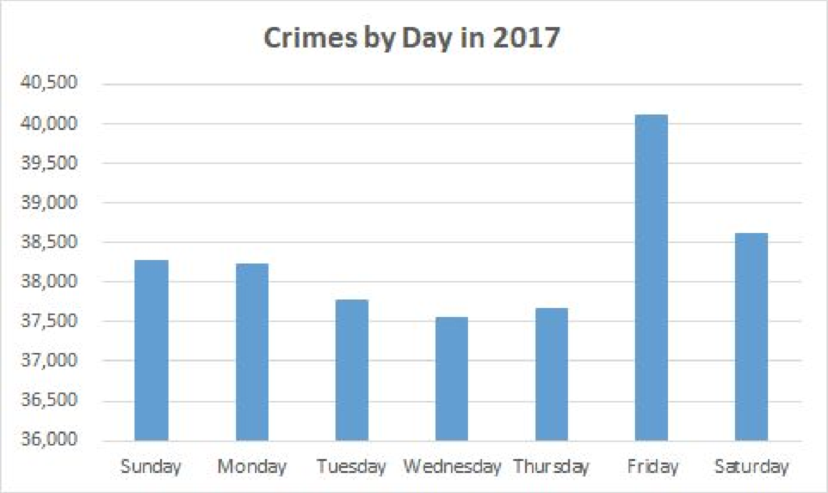
Question 2:

Question two asked for analysis of monthly and daily trends in a specific year (2017) using the extract and count function to obtain the month and day and count the amount of crimes that occured in each. The order by clause ordered the months in chronological order for the reader to understand the seasonality of the crimes.

```
SELECT
EXTRACT(MONTH FROM CRIME_DATE) AS CRIME_MONTH,
COUNT(*) AS CRIMES
FROM CHICAGO
WHERE EXTRACT(YEAR FROM CRIME_DATE) = 2017
GROUP BY  EXTRACT(MONTH FROM CRIME_DATE)
ORDER BY CRIME_MONTH;
```



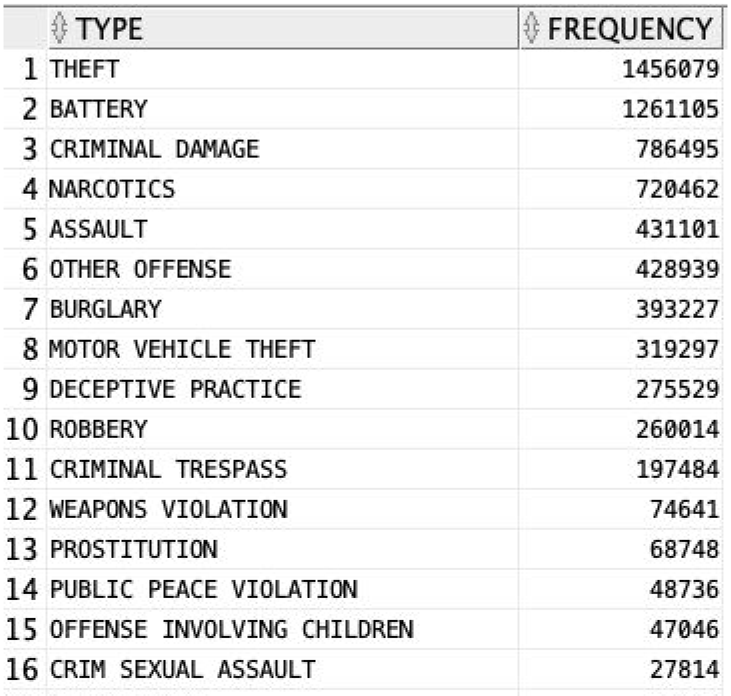
```
SELECT
TO_CHAR(CRIME_DATE, 'd') as CRIME_DAY,
COUNT(*) AS CRIMES
FROM CHICAGO
WHERE EXTRACT(YEAR FROM CRIME_DATE) = 2017
GROUP BY  TO_CHAR(CRIME_DATE, 'd')
ORDER BY CRIME_DAY;
```



Question 3:

For analyzing the crime types, the count, group by and order by functions were used to summarise the most frequent crime type.

```
SELECT TYPE, COUNT(*) AS FREQUENCY
FROM CHICAGO
GROUP BY TYPE
ORDER BY FREQUENCY DESC;
```



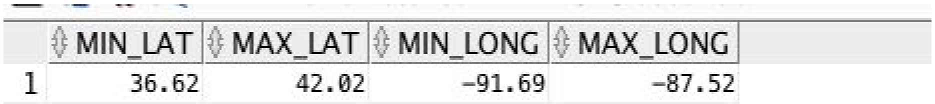
	TYPE	FREQUENCY
1	THEFT	1456079
2	BATTERY	1261105
3	CRIMINAL DAMAGE	786495
4	NARCOTICS	720462
5	ASSAULT	431101
6	OTHER OFFENSE	428939
7	BURGLARY	393227
8	MOTOR VEHICLE THEFT	319297
9	DECEPTIVE PRACTICE	275529
10	ROBBERY	260014
11	CRIMINAL TRESPASS	197484
12	WEAPONS VIOLATION	74641
13	PROSTITUTION	68748
14	PUBLIC PEACE VIOLATION	48736
15	OFFENSE INVOLVING CHILDREN	47046
16	CRIM SEXUAL ASSAULT	27814

Question 4:

To create the 5km by 5km box around Chicago the min and max functions were used to find the min/max for latitude and longitude to find out the bounds of the grid. After finding the min and max, using the formula (x - min) / .045 created a 5km by 5km box around the Chicago area. Finally, using the update function to update the x_coordinate and y_coordinate field to assign each crime a respective box in the grid system.

First find the min and Max range of latitude and longitude of data defining bounding box

SELECT MIN(LATITUDE) AS MIN_LAT, MAX(LATITUDE) AS MAX_LAT,MIN(LONGITUDE) AS MIN_LONG,MAX(LONGITUDE) AS MAX_LONG FROM CHICAGO;



	MIN_LAT	MAX_LAT	MIN_LONG	MAX_LONG
1	36.62	42.02	-91.69	-87.52

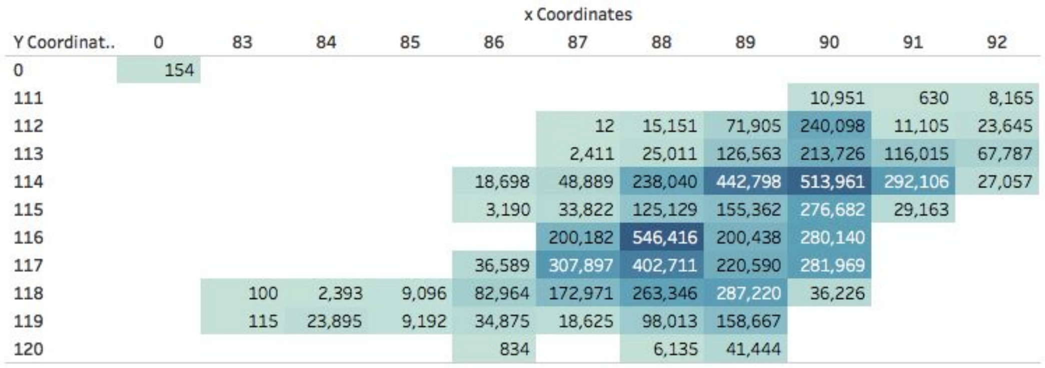
Next, partition the bounding box into equally-sized grids (5km by 5km) and match each crime to a grid cell of x and y coordinates. Finally, adding the coordinates as two new columns in the chicago table.

```
UPDATE CHICAGO
SET X_COORDINATE = FLOOR((LONGITUDE + 91.69)/.045), Y_COORDINATE =
FLOOR((LATITUDE - 36.62)/.045)
WHERE LATITUDE IS NOT NULL;
COMMIT;
```

Question 5:

After creating the grid, a heatmap shows all of the crimes that occurred within it using a simple query grouping by the x_coordinate and y_coordinate that act as an identifier for the boxes. After receiving the output, Tableau was used for visualization.

```
CREATE TABLE CRIME_OVER_GRID
AS (
SELECT X_COORDINATE, Y_COORDINATE, COUNT(*) AS TOTAL_COUNT
FROM CHICAGO
GROUP BY X_COORDINATE, Y_COORDINATE);

SELECT X_COORDINATE, Y_COORDINATE, COUNT(*) AS NUM_CRIMES
FROM CHICAGO
GROUP BY X_COORDINATE, Y_COORDINATE;
```



	x Coordinates										
Y Coordinat..	0	83	84	85	86	87	88	89	90	91	92
0	154										
111									10,951	630	8,165
112						12	15,151	71,905	240,098	11,105	23,645
113						2,411	25,011	126,563	213,726	116,015	67,787
114					18,698	48,889	238,040	442,798	513,961	292,106	27,057
115					3,190	33,822	125,129	155,362	276,682	29,163	
116						200,182	546,416	200,438	280,140		
117					36,589	307,897	402,711	220,590	281,969		
118		100	2,393	9,096	82,964	172,971	263,346	287,220	36,226		
119		115	23,895	9,192	34,875	18,625	98,013	158,667			
120					834		6,135	41,444			

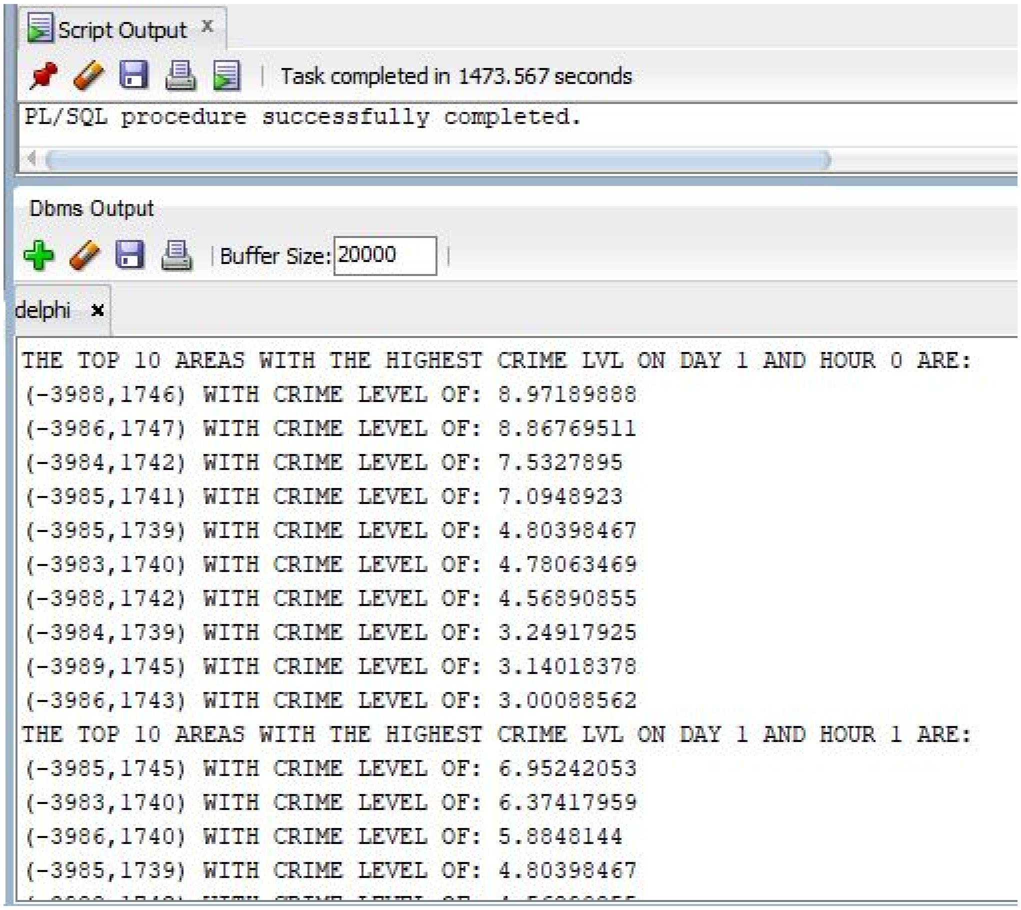
The grid shows that the coordinates 88, 116 have the highest crime count.

Question 7:

The goal of this question was to create a crime monitoring system that found the top 10 grid cells with the most crime activity. The first step was to design a grid table to use as a holder or reference for the grid cell crime activity. Next was to generate a crime table that receives the hourly crimes that would occur using for loops to find every day and every day’s hourly crime. Two for loops was used to simulate looping through every day in a month and every hour in a day. A cursor would select the crimes that occurred in the given day and hour and then insert the crimes into the respective grids in the crime table. Then a query would update the grid table by calculating the total crimes in the previous 24 hours divided by the average daily crime in the respective grid. This was computed by using two queries and inserting the values into two variables. After the grid was updated, a cursor was created that selected the top 10 highest crime levels and the coordinates and would loop and print the coordinates and crime level.

```
DECLARE
V_CRIME_LVL NUMBER;
V_AVG_CRIME_LVL NUMBER;
BEGIN

FOR i in 1..31
LOOP

    FOR j in 0..23
    LOOP

    DECLARE
    CURSOR C_CRIME IS(
    SELECT X_COORDINATE, Y_COORDINATE, CRIME_DATE, SUM(CRIME_LVL)
AS CRIME_LVL
    FROM (
        SELECT X_COORDINATE, Y_COORDINATE, CAST(CRIME_DATE AS DATE)
AS CRIME_DATE,
        CASE
        WHEN ARREST = 'true' THEN 2
        ELSE 1
        END AS CRIME_LVL
        FROM CHICAGO
    WHERE EXTRACT(YEAR FROM CRIME_DATE) = 2002 AND EXTRACT(MONTH
FROM CRIME_DATE) = 1 AND EXTRACT(DAY FROM CRIME_DATE) = i AND
EXTRACT(HOUR FROM CRIME_DATE) = j AND X_COORDINATE IS NOT NULL)
  GROUP BY X_COORDINATE, Y_COORDINATE, CRIME_DATE);
  BEGIN
  FOR V_CRIME IN C_CRIME
  LOOP

    INSERT INTO CRIME VALUES(V_CRIME.X_COORDINATE,
V_CRIME.Y_COORDINATE, V_CRIME.CRIME_DATE, V_CRIME.CRIME_LVL);

    SELECT SUM(CRIME_LVL)
    INTO V_CRIME_LVL
    FROM CRIME
    WHERE X_COORDINATE = V_CRIME.X_COORDINATE AND Y_COORDINATE
= V_CRIME.Y_COORDINATE AND CRIME_DATE BETWEEN
V_CRIME.CRIME_DATE - 1 AND V_CRIME.CRIME_DATE ;

    SELECT  AVG(CRIME_LVL) AS AVG_CRIME_LVL
    INTO V_AVG_CRIME_LVL
    FROM (
      SELECT X_COORDINATE, Y_COORDINATE, CRIME_DATE,
SUM(CRIME_LVL) AS CRIME_LVL
      FROM (
        SELECT X_COORDINATE, Y_COORDINATE, CAST(CRIME_DATE AS
DATE) AS CRIME_DATE,
        CASE
        WHEN ARREST = 'true' THEN 2
        ELSE 1
        END AS CRIME_LVL
        FROM CHICAGO
        WHERE X_COORDINATE = V_CRIME.X_COORDINATE AND
Y_COORDINATE = V_CRIME.Y_COORDINATE)
    GROUP BY X_COORDINATE, Y_COORDINATE, CRIME_DATE)
    GROUP BY X_COORDINATE, Y_COORDINATE;


    UPDATE GRID
    SET CRIME_LVL = (V_CRIME_LVL/V_AVG_CRIME_LVL)
```

```
      WHERE X_COORDINATE = V_CRIME.X_COORDINATE AND Y_COORDINATE
= V_CRIME.Y_COORDINATE;
   END LOOP;
   END;

   DBMS_OUTPUT.PUT_LINE('THE TOP 10 AREAS WITH THE HIGHEST CRIME
LVL ON DAY ' || i || ' AND HOUR ' || j || ' ARE:');
   DECLARE
   CURSOR C_GRID IS (
   SELECT X_COORDINATE, Y_COORDINATE, CRIME_LVL
   FROM(
      SELECT X_COORDINATE, Y_COORDINATE, CAST(CRIME_LVL AS
VARCHAR(10)) AS CRIME_LVL
      FROM GRID
      WHERE CRIME_LVL IS NOT NULL
      ORDER BY CRIME_LVL DESC)
   WHERE ROWNUM <= 10);
   BEGIN
   FOR V_GRID IN C_GRID
   LOOP
      DBMS_OUTPUT.PUT_LINE('(' || V_GRID.X_COORDINATE || ',' ||
V_GRID.Y_COORDINATE || ') ' ||  'WITH CRIME LEVEL OF: ' || V_GRID.CRIME_LVL);
   END LOOP;
   END;

   END LOOP;

END LOOP;
END;
```



```
PL/SQL procedure successfully completed.
```

```
THE TOP 10 AREAS WITH THE HIGHEST CRIME LVL ON DAY 1 AND HOUR 0 ARE:
(-3988,1746) WITH CRIME LEVEL OF: 8.97189888
(-3986,1747) WITH CRIME LEVEL OF: 8.86769511
(-3984,1742) WITH CRIME LEVEL OF: 7.5327895
(-3985,1741) WITH CRIME LEVEL OF: 7.0948923
(-3985,1739) WITH CRIME LEVEL OF: 4.80398467
(-3983,1740) WITH CRIME LEVEL OF: 4.78063469
(-3988,1742) WITH CRIME LEVEL OF: 4.56890855
(-3984,1739) WITH CRIME LEVEL OF: 3.24917925
(-3989,1745) WITH CRIME LEVEL OF: 3.14018378
(-3986,1743) WITH CRIME LEVEL OF: 3.00088562
THE TOP 10 AREAS WITH THE HIGHEST CRIME LVL ON DAY 1 AND HOUR 1 ARE:
(-3985,1745) WITH CRIME LEVEL OF: 6.95242053
(-3983,1740) WITH CRIME LEVEL OF: 6.37417959
(-3986,1740) WITH CRIME LEVEL OF: 5.8848144
(-3985,1739) WITH CRIME LEVEL OF: 4.80398467
```

Conclusion

Leveraging SQL enables analysts to generate insightful statistics on datasets. In this dataset we have found that Chicago’s overall crime rates have decreased since 2001, most crimes are committed on weekends and that most crimes occur in the (88,116) and (90,114) boxes.

In conclusion, this helped us gain a better understanding of the types of crimes committed and location of where they were occured in Chicago.